I. AMENDMENT

In the Claims:

Please amend the claims as follows:

1. (Previously Presented) An image decoder, comprising:

a memory having rows of storage locations; and

a processor coupled to the memory and operable to,

store in a row of the memory intermediate values from first and second matrix columns,

combine the stored intermediate values within the row to generate resulting values for third and fourth matrix columns, and

store the resulting values in respective rows of the memory.

2. (Original) The image decoder of claim 1 wherein the intermediate values comprise Masaki values.

3. (Currently amended) The image decoder of claim 1, wherein:

the memory comprises first, ~~and~~ second, and third rows of storage locations; and

the processor is operable to,

store the intermediate values in the first row of storage locations, ~~and~~

store a first of the resulting values in the second row of storage locations, and

store a second of the resulting values in the third row of storage locations.

4. (Currently amended) The image decoder of claim 1 wherein the processor is operable to generate the intermediate values ~~from the first matrix column~~.

5. (Previously Presented) An image decoder, comprising:

a memory;

a processor coupled to the memory and operable to,

store a column of intermediate values in the memory as a row of intermediate values,

combine the intermediate values within the stored row to generate a column of resulting values, and

store the resulting values in the memory as a row of resulting values;

wherein the intermediate values comprise an even-position even intermediate value, an odd-position even intermediate value, an even-position odd intermediate value, and an odd-position odd intermediate value;

wherein the row has storage locations; and

wherein the processor is operable to store,

the even-position even intermediate value and the even-position odd intermediate value in respective adjacent storage locations, and

the odd-position even intermediate value and the odd-position odd intermediate value in respective adjacent storage locations.

6. (Previously Presented) An image decoder, comprising:

a memory;

a processor coupled to the memory and operable to,

store a column of intermediate values in the memory as a row of intermediate values,

combine the intermediate values within the stored row to generate a column of resulting values, and

store the resulting values in the memory as a row of resulting values;

wherein the intermediate values comprise an even-position even intermediate value, an odd-position even intermediate value, an even-position odd intermediate value, and an odd-position odd intermediate value;

wherein the row has storage locations; and

wherein the processor is operable to store,

the even-position even intermediate value and the even-position odd intermediate value in a first pair of adjacent storage locations, and

the odd-position even intermediate value and the odd-position odd intermediate value in a second pair of adjacent storage locations, the second pair of storage locations being adjacent to the first pair of adjacent storage locations.

7. (Previously Presented) An image decoder, comprising:

a memory;

a processor coupled to the memory and operable to,

store a column of intermediate values in the memory as a row of intermediate values,

combine the intermediate values within the stored row to generate a column of resulting values, and

store the resulting values in the memory as a row of resulting values;

wherein the intermediate values comprise a first even-position even intermediate value, an odd-position even intermediate value, a second even-position even intermediate value, a first even-position odd intermediate value, an odd-position odd intermediate value, and a second even-position odd intermediate value;

wherein the row has storage locations; and

wherein the processor is operable to store,

the first even-position even intermediate value and the first even-position odd intermediate value in a first pair of adjacent storage locations,

the second even-position even intermediate value and the second even-position odd intermediate value in a second pair of adjacent storage locations, the second pair of storage locations being adjacent to the first pair of storage locations, and

the odd-position even intermediate value and the odd-position odd intermediate value in a third pair of adjacent storage locations.

8. (Original) The image decoder of claim 1 wherein the resulting values comprise respective partial inverse-transform values.

9. (Currently Amended) An image decoder, comprising:

memory rows; and

a processor coupled to the memory rows and operable to,

combine a first matrix column of first values with a second matrix column of second values to generate a portion of a resulting matrix column ~~set~~ of resulting values; and

store the ~~set of~~ resulting values in more than one of the memory rows to convert the portion of the resulting matrix column into a corresponding portion of a resulting matrix row.

10. (Previously presented) The image decoder of claim 9 wherein:
the first values comprise even Masaki values; and
the second values comprise odd Masaki values.

11. (Previously presented) The image decoder of claim 9 wherein the processor is further operable to store the first and second values of the first and second matrix columns in more than one of the memory rows.

12. (Previously presented) The image decoder of claim 9 wherein the processor is operable to combine the first column of first values with the second column of second values by adding the first values to the second values.

13. (Previously presented) The image decoder of claim 9 wherein the processor is operable to combine the first matrix column of first values with the second matrix column of second values by subtracting the first values from the second values.

14. (Previously presented) The image decoder of claim 9 wherein the processor is operable to generate the first matrix column of first values and the second matrix column of second values.

15. (Original) An image decoder, comprising:
first and second memory registers having respective storage locations; and
a processor coupled to the registers and operable to,
store each of a set of first intermediate values in every other respective storage location of the first memory register, the set of first intermediate values corresponding to a set of initial values,

store each of a set of second intermediate values in remaining storage locations of the first memory register, the set of second intermediate values corresponding to the set of initial values,

combine each first intermediate value with a second intermediate value that occupies a respective adjacent storage location to generate respective resulting values; and

store each of the resulting values in a respective storage location of the second memory register.

16. (Original) The image decoder of claim 15 wherein:

the first intermediate values comprise even Masaki values; and

the second intermediate values comprise odd Masaki values.

17. (Original) The image decoder of claim 15 wherein the set of initial values comprises a block of discrete-cosine-transform coefficients.

18. (Original) The image decoder of claim 15 wherein:

the set of first intermediate values corresponds to a first subset of the set of initial values; and

the set of second intermediate values corresponds to a second subset of the set of initial values.

19. (Original) The image decoder of claim 15 wherein:

the set of initial values comprises rows of discrete-cosine-transform coefficients;

each of the first intermediate values corresponds to the discrete-cosine-transform coefficients that occupy even locations of a respective row; and

each of the second intermediate values corresponds to the discrete-cosine-transform coefficients that occupy odd locations of the respective row.

20. (Original) The image decoder of claim 15 wherein:

the set of initial values comprises rows of discrete-cosine-transform coefficients, each row having respective even and odd locations;

the processor is operable to generate each of the first intermediate values from the discrete-cosine-transform coefficients that occupy even locations of a respective row; and

the processor is operable to generate each of the second intermediate values from the discrete-cosine-transform coefficients that occupy odd locations of the respective row.

21. (Previously Presented) The image decoder of claim 15 wherein the processor is operable to combine each first intermediate value with a second intermediate value by adding each first intermediate value to the second intermediate value that occupies the respective adjacent storage location.

22. (Previously Presented) The image decoder of claim 15 wherein the processor is operable to combine each first intermediate value with a second intermediate value by subtracting each first intermediate value from the second intermediate value that occupies the respective adjacent storage location.

23. (Previously Presented) The image decoder of claim 15 wherein the processor is operable to:

store each of the set of first intermediate values by storing first and second ones of the first intermediate values in first and third storage locations, respectively, of the first memory register;

store each of the set of second intermediate values by storing first and second ones of the second intermediate values in second and fourth storage locations, respectively, of the first memory register;

combine each first intermediate value with a second intermediate value by combining the first and second ones of the first intermediate values with the first and second ones, respectively, of the second intermediate values to generate respective first and second resulting values; and

store each of the resulting values by storing the first and second resulting values in first and second locations, respectively, of the second memory register.

24. (Previously Presented) The image decoder of claim 15 wherein the processor is operable to:

store each of the set of first intermediate values by storing first and third ones of the first intermediate values in first and third storage locations, respectively, of the first memory register;

store each of the set of second intermediate values by storing first and third ones of the second intermediate values in second and fourth storage locations, respectively, of the first memory register;

combine each first intermediate value with a second intermediate value by combining the first and third ones of the first intermediate values with the first and third ones, respectively, of the second intermediate values to generate respective first and second resulting values; and

store each of the resulting values by storing the first and second resulting values in first and second locations, respectively, of the second memory register.

25. (Original) An image decoder, comprising:

a processor operable to:

receive pixel values that each occupy a respective position within an original row of pixel values,

store the pixel values that respectively occupy every other position of the row in a first continuous section of a register, and

store the pixel values that respectively occupy remaining positions of the row in a second continuous section of the register.

26. (Previously Presented) The image decoder of claim 25 wherein:

the every other row positions comprise even positions within the row; and

the remaining positions comprise odd positions within the row.

27. (Previously Presented) The image decoder of claim 25 wherein the pixel values each comprise a respective encoded pixel value.

28. (Previously Presented) The image decoder of claim 25 wherein the pixel values each comprise a respective discrete-cosine-transform coefficient.

29. (Previously Presented) The image decoder of claim 25 wherein the processor is further operable to receive a block of pixel values, the block including the row of pixel values, the pixel values being arranged in a zigzag pattern.

30. (Previously presented) A method, comprising:

storing in a row of a memory intermediate values from first and second matrix columns;

generating resulting values for third and fourth matrix columns by combining the stored intermediate values within the row; and

storing the resulting values in respective rows of the memory.

31. (Original) The method of claim 30 wherein the intermediate values comprise Masaki values.

32. (Canceled)

33. (Previously Presented) The method of claim 30, further comprising generating the first and second matrix columns of intermediate values.

34. (Previously Presented) A method, comprising:

storing a column of intermediate values as a row of intermediate values;

generating a column of resulting values by combining the intermediate values within the stored row;

storing the resulting values as a row of resulting values;

wherein the intermediate values include an even-position even intermediate value, an odd-position even intermediate value, an even-position odd intermediate value, and an odd-position odd intermediate value; and

wherein the storing the column of intermediate values comprises,

storing the even-position even intermediate value and the even-position odd intermediate value in respective adjacent storage locations of a row, and

storing the odd-position even intermediate value and the odd-position odd intermediate value in other respective adjacent storage locations of the row.

35. (Previously Presented) A method, comprising:

storing a column of intermediate values as a row of intermediate values;

generating a column of resulting values by combining the intermediate values within the stored row;

storing the resulting values as a row of resulting values;

wherein the intermediate values include an even-position even intermediate value, an odd-position even Intermediate value, an even-position odd intermediate value, and an odd-position odd intermediate value; and

wherein the storing the column of intermediate values comprises:

storing the even-position even intermediate value and the even-position odd intermediate value in a first pair of adjacent storage locations, and

storing the odd-position even intermediate value and the odd-position odd intermediate value in a second pair of adjacent storage locations, the second pair of storage locations being adjacent to the first pair of adjacent storage locations.

36. (Previously Presented) A method, comprising:

storing a column of intermediate values as a row of intermediate values;

generating a column of resulting values by combining the intermediate values within the stored row;

storing the resulting values as a row of resulting values

wherein the intermediate values include a first even-position even intermediate value, an odd-position even intermediate value, a second even-position even intermediate value, a first even-position odd Intermediate value, an odd-position odd intermediate value, and a second even-position odd intermediate value; and

wherein the storing the column of intermediate values comprises,

storing the first even-position even intermediate value and the first even-position odd intermediate value in a first pair of adjacent storage locations,

storing the second even-position even intermediate value and the second even-position odd intermediate value in a second pair of adjacent storage locations, the second pair of storage locations being adjacent to the first pair of storage locations, and

storing the odd-position even intermediate value and the odd-position odd intermediate value in a third pair of adjacent storage locations.

37. (Currently Amended) A method, comprising:

combining a first matrix column of first values with a second matrix column of second values to generate a portion of a resulting matrix column ~~set~~ of resulting values; and

storing the ~~set of~~ resulting values in more than one row of a memory to convert the portion of the resulting matrix into a corresponding portion of a resulting matrix row.

38. (Previously presented) The method of claim 37 wherein:
the first values comprise even Masaki values; and
the second values comprise odd Masaki values.

39. (Previously presented) The method of claim 37, further comprising storing the first and second values of the first and second matrix columns in more than one row of the memory.

40. (Previously presented) The method of claim 37 wherein the combining comprises adding the first values to the second values.

41. (Previously presented) The method of claim 37 wherein the combining comprises subtracting the first values from the second values.

42. (Previously presented) The method of claim 37, further comprising:
generating the first matrix column of first values; and
generating the second matrix column of second values,

43. (Original) A method, comprising:

storing each of a set of first intermediate values in every other respective storage location of a first memory register, the set of first intermediate values corresponding to a set of initial values;

storing each of a set of second intermediate values in remaining storage locations of the first memory register, the set of second intermediate values corresponding to the set of initial values;

generating respective resulting values by combining each first intermediate value with a second intermediate value that occupies a respective adjacent storage location of the first memory register; and

storing each of the resulting values in a respective storage location of a second memory register.

44. (Original) The method of claim 43 wherein:

the first intermediate values comprise even Masaki values; and

the second intermediate values comprise odd Masaki values.

45. (Original) The method of claim 43 wherein the set of initial values comprises a block of discrete-cosine-transform coefficients.

46. (Original) The method of claim 43 wherein:

the set of first intermediate values corresponds to a first subset of the set of initial values; and

the set of second intermediate values corresponds to a second subset of the set of initial values.

47. (Original) The method of claim 43 wherein:

the set of initial values comprises rows of discrete-cosine-transform coefficients;

each of the first intermediate values corresponds to the discrete-cosine-transform coefficients that occupy even locations of a respective row; and

each of the second intermediate values corresponds to the discrete-cosine-transform coefficients that occupy odd locations of the respective row.

48. (Original) The method of claim 43, further comprising:

wherein the set of initial values comprises rows of discrete-cosine-transform coefficients, each row having respective even and odd locations;

generating each of the first intermediate values from the discrete-cosine-transform coefficients that occupy even locations of a respective row; and

generating each of the second intermediate values from the discrete-cosine-transform coefficients that occupy odd locations of the respective row.

49. (Original) The method of claim 43 wherein the generating comprises adding each first intermediate value to the second intermediate value that occupies the respective adjacent storage location of the first memory register.

50. (Original) The method of claim 43 wherein the generating comprises subtracting each first intermediate value from the second intermediate value that occupies the respective adjacent storage location of the first memory register.

51. (Original) The method of claim 43 wherein:

the storing the first intermediate values comprises storing first and second ones of the first intermediate values in first and third storage locations, respectively, of the first memory register;

the storing the second intermediate values comprises storing first and second ones of the second intermediate values in second and fourth storage locations, respectively, of the first memory register;

the generating comprises combining the first and second ones of the first intermediate values with the first and second ones, respectively, of the second intermediate values to generate respective first and second resulting values; and

the storing the resulting values comprises storing the first and second resulting values in first and second locations, respectively, of the second memory register.

52. (Original) The method of claim 43 wherein:

the storing the first intermediate values comprises storing first and third ones of the first intermediate values in first and third storage locations, respectively, of the first memory register;

the storing the second intermediate values comprises storing first and third ones of the second intermediate values in second and fourth storage locations, respectively, of the first memory register;

the generating comprises combining the first and third ones of the first intermediate values with the first and third ones, respectively, of the second intermediate values to generate respective first and second resulting values; and

the storing the resulting values comprises storing the first and second resulting values in first and second locations, respectively, of the second memory register.

53. (Original) A method, comprising:

storing the pixel values that respectively occupy every other position of a row of pixel values in a first continuous section of a register; and

storing the pixel values that respectively occupy remaining positions of the row in a second continuous section of the register.

54. (Original) The method of claim 53 wherein:

the every other row positions comprise even positions within the row; and

the remaining positions comprise odd positions within the row.

55. (Original) The method of claim 53 wherein the pixel values each comprise a respective encoded pixel value.

56. (Original) The method of claim 53 wherein the pixel values each comprise a respective discrete-cosine-transform coefficient.

57. (Original) The method of claim 53, further comprising extracting the row of pixel values from a zigzag-encoded block of pixel values.

58. (Previously presented) The image decoder of claim 9 wherein the resulting values comprise inverse-discrete-cosine-transform values.

59. (Previously presented) The method of claim 37 wherein the resulting values comprise inverse-discrete-cosine-transform values.

60. (New) The image decoder of claim 9 wherein an arrangement of the resulting values within the resulting matrix row is different than an arrangement of the resulting values within the resulting matrix column.

61. (New) The image decoder of claim 9 wherein an arrangement of the resulting values within the resulting matrix row is the same as an arrangement of the resulting values within the resulting matrix column.

62. (New) The image decoder of claim 9 wherein each of the memory rows comprises a single respective register.

63. (New) The image decoder of claim 9 wherein:
the portion of the resulting matrix column comprises the entire matrix column; and
the portion of the resulting matrix row comprises the entire matrix row.